25

Data Mining for Problem Discovery

Donald E. Brown
University of Virginia
U.S.A.

1. Introduction

Data mining typically focuses on knowledge discovery. This means the identification or recognition of persistent patterns or relationships in data. Data mining can also support problem discovery or the identification of patterns or relationships in data that represent either causal mechanisms or association mechanisms. Association mechanisms fall short of causality but can provide useful insights for the design of solutions in the problem domain. A common goal of problem discovery is to identify the causal or association mechanisms behind metrics that measure system performance or behavior. Depending on the domain these metrics can be quantitative, e.g., cost of operation, or qualitative, e.g., acceptable or unacceptable behavior.

Data mining contains many approaches that can support problem discovery. This chapter reviews some significant examples and shows how their combination provides useful results. Before reviewing these approaches we note that problem discovery places four key requirements on the data mining approaches. The first is for unsupervised learning techniques for data association. Data association derives from unsupervised learning techniques that find structure in data. As such, data association seeks patterns of domain specific similarity among observations and uses a variety of similarity measures to find these patterns.

A second and closely related requirement for data association for problem discovery is the need for text association. Much of problem discovery concerns finding relationships in free text as well as fixed field data. Free text presents many challenges and a number of data mining techniques have been proposed to group documents and identify similarities. Problem discovery can exploit these methods but requires that they work closely with data association discoveries made using the fixed field data. The combination of free text and fixed field data can provide considerable information about the underlying causal or association mechanisms at the heart of problem discovery.

A third requirement for problem discovery methods applies to the use of supervised learning techniques. Specifically these techniques must produce interpretable results. This means that the discovery methods must reveal insights into causal or association mechanisms that contribute to the problem. So, unlike traditional data mining, problem discovery focuses more on interpretability at the possible expense of accuracy.

Finally, problem discovery requires the integration of methods from both supervised and unsupervised learning. By definition the exact nature of the problem is unknown so the application of, say, supervised learning tends to provide a narrow focus that misses important aspects of the problem. In contrast, unsupervised learning provides too broad a perspective in the presence of known instances of problematic behavior. Hence, problem discovery requires

integrated strategies that combine results from supervised and unsupervised learning approaches.

The organization of this chapter provides a pathway for showing how we can meet these four requirements for problem discovery methods. The chapter begins with two sections dedicated to the current data mining techniques with most direct applicability to problem discovery. The next section, Section 2, reviews relevant results from unsupervised learning and the section following that, Section 3, provides the background in supervised learning techniques. Both sections show the strengths and weaknesses of these techniques for the specific issues in problem discovery. After this foundation, Section 4 shows how we can extend existing methods and integrate them into an approach for problem discovery. Finally, Section 5 provides an example of the use of the problem discovery methods for uncovering factors to guide strategies to reduce the number and severity train accidents.

2. Unsupervised learning methods for problem discovery

Problem discovery typically begins with the application of methods from unsupervised learning. Unsupervised learning techniques find patterns in data where the variables in the data do not include any response or output variables. Even in data sets that have output variables, the use unsupervised methods provides insight into the relationships among the variables needed to discover lurking or hidden problems not visible by simply apply supervised learning techniques.

A major difficulty with unsupervised learning follows from the lack of one or more output variables; namely, these methods do not have strong evaluation metrics. The presence of one or more output variables in the case of supervised learning means that we can measure the deviation from the actual to the predicted output and score the methods the accordingly. Since the data for unsupervised learning techniques do not contain output variables, we do not have the same straightforward measure of effectiveness. Hence, we typically judge unsupervised learning with a variety of subjective measures. This has led to a wide variety of methods and this section cannot possibly provide coverage of them all. Instead, we focus on those methods with the most direct applicability to problem discovery: association methods. Subsection 2.1 describes association rules, Subsection 2.2 overviews methods for associating variables, and Subsection2.3 gives an introduction to clustering. Lastly, Section 2.4 describes current methods for text mining that have applicability to problem discovery.

2.1 Association rules

Association rules are actually part of a collection of data mining techniques known as market basket analysis. Market basket analysis seeks to organize data on customer purchase behavior. Consider, for example, data on the purchase of items by customers at a store over a recent period of time. Do these customers frequently buy the same groups of items? So, for example, when they purchase cheese, do they also purchase wine? Understanding these associations may help store managers to better inventory, display, and manage their marketable items. Despite the name, market basket analysis provides useful methods for domains outside of retail sales. For instance, in health care, market basket analysis can provide an understanding of associations among patients with demands for similar services and treatments. In this sense the market basket contains a group of services purchased or requested by the customer.

Consider the set of all possible items or services that can be placed in a customer's market basket. Then each item has value associated with it which represents the quantity purchased by that customer. The goal of market basket analysis is to find those values of items for which

their joint probability of occurrence is high. Unfortunately, for even modest sized businesses this problem is intractable.

Instead, analysts typically simplify the problem to allow only binary values for the items or services. These values reflect a yes or no decision for that item and not the quantity. Each basket then is represented as a vector of binary valued variables. These vectors show the associations among the items. The results are typically formed into association rules. For example, 'customers who buy cheese (c) and bread (b) also buy wine (w)' is converted to the rule,

$$c, b \Rightarrow w \tag{1}$$

These rules are augmented by the data to show the support and the confidence in the rule. Support for a rule means the proportion of observations or transactions in which both items occurred together. In the example in 1 the support for rule indicates the proportion of purchases in which cheese, bread, and wine appear together. The confidence for a rule shows the proportion of times the consequent of the rule occurs within the set of transactions containing the antecedent. In the above example, the confidence for the rule would be proportion of times that wine was purchased among those customers who also purchased cheese and bread.

A number of algorithms have been developed to find rules of this sort. One of the earliest and most commonly used of these algorithms is the Apriori algorithm Agrawal et al. (1996). Other algorithms for association rules have been developed and Zheng et al. (2001) provides comparison of several of these algorithms. For purposes of this chapter the Apriori algorithm provides a good illustration of the usefulness of association rule techniques for problem discovery.

The Apriori algorithm operates on sets of items in baskets, i.e., those with value one in binary formulation. These sets are called itemsets. The algorithm begins with the most frequently observed single itemsets. This means those items most often purchased by themselves. The algorithm uses these sets to find the most commonly purchased 2 item itemsets. At each iteration it prunes itemsets that do not pass a threshold on support or the frequency with which the itemset appears in the transactions. Once the common 2 item itemsets are found that pass this threshold, the algorithm uses these to consider 3 item itemsets. These are again pruned based on the support threshold. The algorithm proceeds in this way and stops when the threshold test is not satisfied by any itemset.

The Apriori algorithm and other association rule algorithms produce rules of the type shown in 1 with both confidence and support values. For problem discovery these results can provide some insight into association and causal mechanisms. For instance, in trying to determine the problems underlying train accidents we would be interested in association rules that show relationships between potential causes of accidents and measures of accident severity. Unfortunately, current versions of these algorithms cannot handle non-binary variables. This severely restricts the usefulness of association rules for problem discovery.

2.2 Variable association

Like association rules, variable association methods look for patterns among the variables in the data set. However, unlike association rules, the more general methods of variable association attempt to find simple, typically linear, relationships among the variables without the additional requirement of finding a rule that represents that relationship. By relaxing this

latter requirement variable association can work with the non-binary data typically found in problem discovery.

A common and effective method for associating variables is principal components. Principal components provide linear combinations of the variables that can approximate the original data with fewer dimensions. The linear combinations found by principal components satisfy the following properties:

1. The variance of each principal component is maximized;
2. The principal components are pairwise orthogonal; and
3. Each component is normalized to unit length.

The first property follows from a desire to maintain as much of the spread of the original data set as possible. The second property is for convenience. Orthogonality means that the projection into the of the principal components is a projection of the original basis functions. The final property is also for convenience. In this case it enables obtainment of a bounded solution to the optimization problem.

With these properties it is straightforward to find the linear combinations of the variables that produce the principal components. Let X be the data matrix with N rows and p columns or variables. Let S be the $p \times p$ variance-covariance matrix for this data matrix. For principal components these variables must be in Euclidean space. Now consider U the matrix of principal components with p rows and q columns, where $q \leq p$. For the first principal component, u_1, the properties above mean that we want to find the

$$\underset{u_1,\lambda_1}{\operatorname{argmax}}\{u_1^T S u_1 + \lambda_1(1 - u_1^T u_1)\}. \tag{2}$$

The solution to 2 is given by

$$S u_1 = \lambda_1 u_1. \tag{3}$$

The solution in 3 implies that the first principal component is the eigenvector of the variance-covariance matrix, S with the largest eigenvalue, λ_1. The remaining principal are similarly defined and are orthogonal to all preceding principal components. Hence, U is the matrix of eigenvectors for S and $\lambda_1, \ldots, \lambda_q$ are the eigenvectors. The eigenvalues also provide the variance of the data in the projection of respective principal component.

In data mining we typically look for solutions in which the number of principal components is less than the number of variables in the data set, i.e., $q < p$. The proportion of variance in the data in a subset of the principal components is found from the appropriate ratio of eigenvalues. For example, the variance of the data projected into the first two principal components is $(\lambda_1 + \lambda_2)/(\lambda_1 + \lambda_2 + \ldots + \lambda_K)$.

Since data mining looks for associations among variables, the results are particularly interesting when a small number of principal components explains a large amount of the variance in the database. It is unrealistic to expect a small number of variables to explain nearly all of the variance; however, it is often possible to find a small number of principal components that explain as much as half of the original variance. As the number of variables gets larger it can become harder to achieve this goal.

A method closely related to principal components is singular valued decomposition (SVD). To see the relationship, again consider a data set, X, with N observations and p variables. We can decompose X, into 2 orthogonal matrices and a diagonal matrix Golub & Loan (1983) defined as follows:

$$X = UDT^T \tag{4}$$

In 4 U is an $N \times p$ matrix called the left singular vectors and T is $p \times p$ matrix called the right singular vectors. Also $U^T U = I$ and $T^T T = I$. D is diagonal matrix with dimensions and $p \times p$ and whose elements are the singular values. The major advantage to this decomposition is that it enables variable association for problems in which $N < p$.

Principal components has seen a number of important extensions. Among these are variational Bishop (1999b) and Bayesian Bishop (1999a) methods for principal components. In addition to principal components many other methods exist for associating variables. Some representative methods include partial least squares Wold (1975), ridge regression Hoerl & Kennard (1964), and independent components Comon (1994). A discussion and comparison of methods can be found in Copas (1983).

While these variable association methods provide a mechanism to link variables for problem discovery, they work only on quantitative variables. Much of the data in problem discovery consists of categorical variables and text. Hence, these methods cannot effectively provide complete solutions for problem discovery.

2.3 Clustering

Clustering is another class of unsupervised learning techniques with applicability to problem discovery. Quite simply the goal of clustering is to organize or group items based on the properties of those items. This goal has been of practical concern for a long time; hence, there exists a large number of approaches to this problem. Modern clustering techniques have their roots in statistics and taxonomy and these areas provide the foundation for many of the data mining techniques.

Clustering begins with a distance, similarity, or dissimilarity score for pairs of observations. The most common distances are Euclidean, Manhattan (or city block), and Max. Suppose we have observations x_i, x_j each consisting vectors of quantitative values over p variables. Then the Euclidean, Manhattan, and Max distances are defined as follows:

$$\begin{aligned} d_{\text{Euclid}} &= \left[\sum_{k=1}^{p} (x_{ik} - x_{jk})^2\right]^{\frac{1}{2}} \\ d_{\text{Man}} &= \sum_{k=1}^{p} |x_{ik} - x_{jk}| \\ d_{\text{Max}} &= \max_k \{|x_{ik} - x_{jk}|\}. \end{aligned}$$

A commonly used similarity is cosine, defined as $x_i \cdot x_j$.

Clustering algorithms employ the distance, similarity, or dissimilarity scores to group observations. For convenience researchers often categorize clustering algorithms as hierarchical, partitioning, or model-based, although these categories are neither inclusive nor mutually exclusive. We briefly describe these approaches to show their applicability to problem discovery.

As the name implies hierarchical clustering provides a level that shows the point of formation of different clusters. This allows for viewing of the data set in two dimensions: one (typically the abscissa) showing the cluster labels and the other (the ordinate) showing the level of

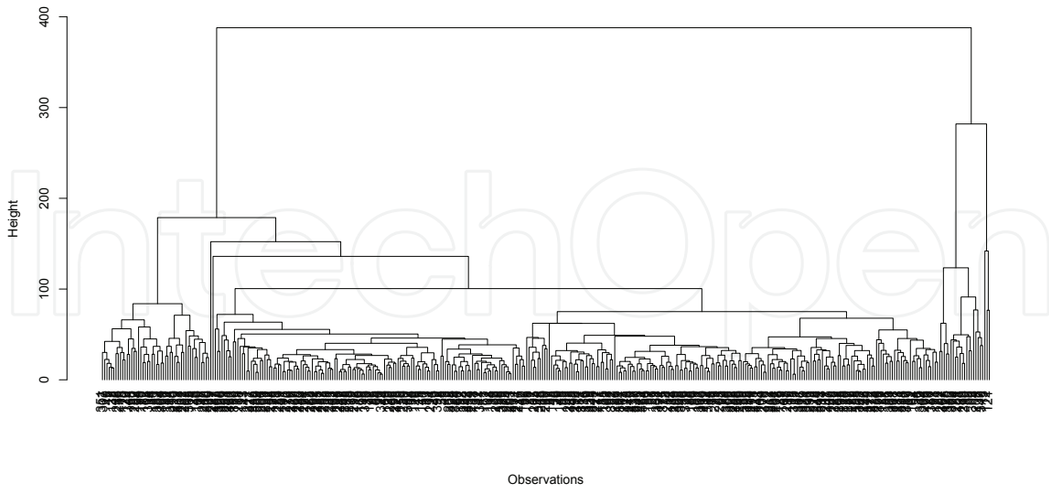


Fig. 1. Dendrogram of average link clustering

cluster formation. This plot is called a dendrogram (illustrated in Figures 1). The combination of labels and levels provides an indication of the patterns and structures in the database.

Partitioning methods group the observations based on their distance, similarity, or dissimilarity scores with each other. K-means is a typical and commonly used partitioning method. K-means requires the a-priori choice of the number of clusters and then randomly assigns observations to clusters. The algorithm next calculates the cluster centroids from this initial clustering. In the subsequent step the algorithm moves observations that have smaller dissimilarity or greater similarity with other centroids than they have to their assigned centroid. New centroids are calculated after this reassignment step. Once again observations are moved if their minimum dissimilarity or maximum similarity is with a centroid different from their assigned centroid. The process stops when no reassignments are made. It is easy to show this algorithm converges in a finite number of steps.

A number of researchers have extended the basic K-means formulation to improve its performance over a wide range of problems. A major disadvantage to K-means and its extensions for problem discovery is that it produces clusters of roughly the same size. Problem discovery tends to have unbalanced clusters with widely different sizes. Also, K-means requires knowledge of the number of clusters which requires the further use of various forward and backward search strategies.

A simple partitioning approach variously called the leader algorithm or nearest neighbor clustering starts by putting the first observation in the first cluster. The algorithm next finds the similarity or dissimilarity of the second observation with the first cluster, $s(x_2,c_1)$ or $s(x_2,c_1)$, respectively. If $s(x_2,c_1) > \tau$ or $d(x_2,c_1) < \tau$ then the second observation is added to the first cluster. Otherwise a new cluster is formed out the second observation. This logic is used to cluster the remaining observations. Typically the similarity or distance between the new observation and the clusters are found from the maximum similarity or minimum distance between the new observation and all observations assigned to the cluster. However, the similarity or distance could also be the maximum or the average of the points in the cluster, or another suitable choice function.

This algorithm is very efficient ($o(N)$), but is also order dependent. While it does not require

the explicit delineation of the number of clusters it does require specification of the threshold, τ and that indirectly specifies the number of clusters.
Model-based clustering uses a probabilistic model for the data. This model assumes the data come from draws against a mixture distribution with k components. One popular choice uses mixtures of Gaussians so the distribution of an observation x_j is found as

$$p(x_j) = \sum_{i=1}^{k} \pi_i \phi(x_j|\mu_i, \Sigma_i) \tag{5}$$

where $\phi(x|\mu,\Sigma)$ is a Gaussian distribution with parameters μ and Σ, and π_i are mixing coefficients with $\pi_i \in [0,1]$ and $\sum_{i=1}^{k} \pi_i = 1$. Unlike partitioning methods model-based clustering makes probabilistic assignments of observations to clusters.
Model-based clustering uses the expectation-maximization (EM) algorithm to find the parameters, μ_i, Σ_i, π_i and k for $i = 1,\ldots,k$. This algorithm proceeds in a fashion similar to k-means. The EM algorithm's first step initializes all the parameters (either randomly or according to some specified values). Next the algorithm finds $Pr(x_j \in C_i)$, the probability, x_i was drawn from cluster C_i for all observations, $j = 1,\ldots,N$ and clusters, $j = 1,\ldots,k$. The EM algorithm next re-estimates the parameters that maximize the likelihood of the current cluster assignments. The $Pr(x_j \in C_i)$ are calculated again for these new parameter estimates. The algorithm proceeds in this way and stops when no new assignments are made.

2.4 Text mining

Text mining does not fit entirely within unsupervised learning. However, as indicated in Section 1 text association is a critical component of problem discovery and methods from text mining or text data mining provide a foundation for meeting this requirement. At its most general level, text mining is a process of deriving consistent patterns from text. Text mining first structures the input text by parsing narrative data, then derives patterns within the structured data, and finally evaluates and interprets the output. The first step in text mining is similar to transforming free text into feature vectors in information retrieval. However, text mining usually applies more techniques on the structured data to derive useful information and speed this process. Typically, text mining tasks include information extraction, text categorization, summarization, and clustering Konchady (2006).
Information extraction techniques extract interesting information from the text. For example, they can extract peoples names, locations, vehicle types, and accidents from a passage. Information extraction techniques can be rule-based Ciravegna et al. (1999) and Krupka & Hausman (1998), statistics-based Witten et al. (1999), or use machine learning Baluja et al. (1999). Text categorization and clustering are like categorization and clustering performed in data mining, but performed on narratives or text. Applications of text categorization are described by Fall et al. (2003) and Gentili et al. (2001) and text clustering algorithms are described by Deerwester et al. (1990) and Hotho et al. (2001). Text summarization techniques seek to automatically summarize passages or narratives. These techniques can be based on linguistic rules, statistics, or both. Text summarization algorithms are described by Mani & Maybury (1999).
Other text mining techniques are developed to process text for specific applications. Yetisgen-Yildiz and Pratt Yetisgen-Yildiz & Pratt (2006) developed a literature-based discovery system called LitLinker to mine the biomedical literature for new, potentially interesting connections between biomedical terms. To reduce the dimensions of word vectors, they used Medical Subject Headings (MeSH) keywords assigned to the documents to capture the

content of the documents. The system uses a MeSH dictionary which is manual built by experts and the resulting text mining system can process narrative information quickly. Corley and Mihalcea Corley & Mihalcea (2005) presented a knowledge-based method for measuring the semantic-similarity of texts. They introduced a text-to-text semantic similarity metric by combining metrics of word-to-word similarity and language models. The word-to-word similarity metrics measure the semantic similarity of words using semantic networks or distributional similarity learned from large text collections. The language models provide the specificity of words. In their method, they determined the specificity of a word using the inverse document frequency as discussed in section 2.2. The specificity of each word is derived from the British National Corpus. Similarity between texts is determined by word-to-word similarities between all the words in the texts and specificity of each word. Experiments show their method outperforms the traditional text similarity metrics based on lexical matching. Hoang Hoang (2004) presented a method using the principal components to reduce dimensions of word vectors to reduce the time of text mining. paper discussed how the principal components method is used in information retrieval and how the latent semantic indexing is related to the principal component method. With word vectors from texts, the proposed method computes the principal components for these vectors and uses the reduced dimension vectors to represent the texts. Experiments showed the method works efficiently as well as effectively.

3. Supervised learning methods for problem discovery

Problem discovery requires techniques that go beyond discovering relationships between variables and observations through unsupervised learning. We also require techniques that can further characterize relationships between variables and can indicate the importance of the variables in these relationships. Supervised learning provides a set of techniques for accomplishing these tasks.

The inputs to supervised learning contain a further segmentation of the variable types into predictors and response. The goal of supervised learning is to find the relationships between the predictor variables and the response variables that will enable accurate and ideally fast estimation of the response values.

Predictor variables are further decomposed into control and environmental variables. The values of control variables can be set by the users or systems operating in the problem domain. Environmental variables are exogenous to these users and systems and hence cannot be set by them. Response variables are the outputs of the processes or systems in the problem domain. If an unsupervised learning technique works well it will produce a function that accurately maps the heretofore unseen predictor variable inputs to accurate estimates of the response variables.

As with unsupervised learning, the area of supervised learning encompasses a large number of techniques. Again we focus on the major techniques applicable to problem discovery. Also, to keep the notation manageable we describe these techniques using only a single response variable. The extension to the multi-variate case is conceptually straightforward once the univariate case is understood. The section begins with numeric response, since this builds directly on commonly used regression or least squares techniques. From there the discussion moves to the categorical response variables. Most data mining methods can handle both types of response, although the actual mechanics of the methods change with changing response type.

Unlike unsupervised learning, supervised learning has direct methods for measuring and

evaluating performance or accuracy. Section 3.3 describes the fundamentals of evaluating the accuracy of supervised learning techniques.

While accuracy is important in may applications of supervised learning for problem discovery, interpretability or understanding the contribution of the variables to the response is also important. Not all data mining methods are easily interpretable. Among the most interpretable are tree-based methods. Among the least interpretable are support vector machines and other kernel methods. Since problem discovery depends on interpretability this section will describe only those methods with good interpretability that have application to problem discovery.

3.1 Regression

As noted the mechanics of supervised learning methods changes with the response variable . Numeric response variables have values over a continuum or a reasonably large set of integers. Categorical response variables have values that are unordered labels, such as names. Some response values are simply ordered which means they do not fit neatly into either of the previous categories. For the purposes of this chapter, the methods that can handle categorical response can also handle ordered response, although not necessarily in a manner that fully exploits the ordering.

For numeric response variables the field of statistics provides a rich set of techniques that fall under the rubric regression. Many of these regression methods find a linear function of the predictor variables that minimizes the sum of square differences with the response values. Let y_i be the response value and x_i be the vector of predictor values for observation i, $i = 1,\ldots,n$. Also let f be the function that estimates the response and θ be the vector of parameters in this function. For a given functional form, least squares chooses the parameters that minimize the sum of square distances to each observed response value. So, the estimated parameters, $\hat{\theta}$ are given by

$$\hat{\theta} = argmin\{\sum_{i=1}^{n}(y_i - f(\theta, x_n))^2\}. \tag{6}$$

A convenient choice for f in equation (6) is a linear form and for p predictor variables this gives the following:

$$f(\theta, x_n) = \theta_0 + \theta_1 x_1 + \cdots + \theta_p x_p. \tag{7}$$

The linear form show in equation (7) is useful for interpretation. Each coefficients on variables, $\theta_i, i = 1,\ldots,p$, provides an easy interpretation as the change in response for a one unit change the variable while holding all other variables constant. Of course, holding all other variables constant is typically a mental exercise, since only in controlled experiments can we meet this condition. Also, as the relationship becomes nonlinear the coefficients provide less easily understood interpretations.

Other measures of interpretability provided by regression models are the statistics obtained for the model and for each coefficient in the model. For the model the statistic is a value for the F distribution and for the individual coefficients the statistic is a value from the t distribution, although the F distribution can also be used. These distributions follow directly from the sum of squares errors where the errors measure the absolute difference between the regression predictions and the actual values. Details of this can be found in Seber (1984). These F and t values allow tests of hypotheses, such as, $\theta_1 = \theta_2 = \ldots = \theta_p = 0$ for the model and $\theta_i = 0$ for the

coefficients, $i = 1,\ldots,p$. The F statistic can also test hypotheses about groups of coefficients, e.g., $\theta_i = \theta_j = \theta_k = 0$ These tests provide a measure of variable importance in the model.
Least squares regression as described here is the topic of a vast literature, for example, see Seber (1984). It has also extended numerous ways to include handling of correlation among variables Hoerl & Kennard (1964) and correlation among observations Kedem & Fokianos (2003).
While least squares regression models data mining problems with numeric response variables, to find patterns with categorical response variables requires a different approach to regression. Consider the simplest case where the categorical variable is binary, e.g., the accident had deaths or no deaths. Least squares regression would not be appropriate for this problem since it would provide predictions that would lie outside the binary response values.
An extension to the regression approach is accomplished by modeling the probability of a binary response. With n independent observations then the probability of k occurrences of an event is given a binomial distribution. Let π be the parameter for this binomial distribution which is simply the probability of an event in any observation. A convenient, but by no means unique model, assumes this probability, π is a logistic function of the predictors with parametric vector θ. This yields the following:

$$log[\frac{\pi}{1-\pi}] = \theta^T x \tag{8}$$

where $x^T = (x_0, x_1, \ldots, x_k)$ and $\theta^T = (\theta_0, \theta_1, \ldots, \theta_k)$.
As with linear regression, logistic regression provides insight into influence of the predictors on the response. Now instead of using the F and t distributions, we use a χ^2 distribution as a large sample distribution for the likelihood ratio. This allows for the same hypotheses tests as we used for interpretability in linear regression, i.e., $\theta_1 = \theta_2 = \ldots = \theta_p = 0$ for the model and $\theta_i = 0$ for the coefficients, $i = 1,\ldots,p$. The coefficients themselves show the factor by which the odds ratio in equation (8) changes as the result of a one unit change in the variable while holding all other variables constant. Again, this interpretation is easy for linear models, such as equation (8), but not easy for nonlinear models. The motivates interest in other techniques that provide interpretability across more complex relationships.

3.2 Tree-based methods

Tree-based methods provide models for both numeric and categorical response variables. The advantage tree-based methods have over other supervised learning approaches is their interpretability. At the foundation all tree-based methods is the construction of a tree that represents a partition of the data set into regions for which a particular response value is prominent. The partitioning is accomplished through a series of questions. For example, at the time of the accident was the vehicle traveling at a speed in excess of posted maximum? Observations with affirmative answers to this question are separated from those with negative answers. Additional questions continue the partitioning until regions are found that primarily contain a single response value for categorical response variables or are near a value for a numeric response.
To see how this partitioning can be viewed as a tree, let each node represent a question that partitions the data. The answer to one question, leads to another question (the branch of the tree) until we finally arrive at the leaf. The leaf nodes give the estimated classification for a categorical response or value for a numeric response. This combination of questions or nodes and questions that follow questions can be represented as a tree (although one that is growing down rather than up).

The resulting tree is easily interpretable since it is simply a set of linked questions. This reasoning is familiar to most people and hence the output tree-based methods can be implemented in virtually all settings with little explanation. For problem discovery this interpretability means uncovering relations that might not otherwise be exposed amidst. Unlike regression methods, trees can display nonlinear relationships in a form that is easy to interpret. Obviously a large tree with many variables becomes less easily understood, but even in these cases it is possible to view the tree in segments or branches. These branches can aid in understanding and problem discovery.

We can construct tree classifiers and regression trees with a variety of algorithms. One of the most effective of these, known as recursive partitioning (RP), was developed by Breiman et al. (1984). This algorithm constructs trees by providing answers to three tree construction questions: (1) When to stop growing the tree; (2) What label to put on a leaf node; and (3) How to choose a question at a node.

The question, when to stop growing the tree, they answered simply by not stopping. Instead the RP algorithm grows the tree out to its maximum size (e.g., each observation in its own terminal node). RP then prunes the tree back to a size that best predicts a set of hold-out samples (the actual approach used is discussed in Section 3.3). This pruning approach avoids generating trees that are not effective because they did not consider a sufficiently large and cooperative set of nodes.

The second question, what label to put on a leaf node, has an easy answer: for categorical response choose the category with the most members in the node; and for numeric response take the average or median. Ties among categories are simply reported. This approach means that the algorithm provides a quick estimate of the probabilities for each category in the leaf nodes. It also provides an empirical distribution for numeric values in the leaf nodes.

The third question, How to choose a question at a node, has a more involved answer. RP develops a question for a node by considering the values of every variable for every observation in a node as possible question. For numeric variables the questions considered ask if the variable has a value less than the mid point between two adjacent values of that variable for the observations in the node. For categorical variables, the questions ask if the value of the variable is a member of one of the proper subsets of the values observed for that variable in the node's observations. The algorithm chooses from this large set the question that best partitions the data. Best is measured by purity of the results (see Breiman et al. (1984) for definitions of purity). So, for example, a question that partitions the data into nodes with dominant class labels is preferred to one that has the labels in roughly equal proportions. Similarly, a regression tree that partitions the data into nodes whose response values have low variance is preferred to one one high variance.

Other approaches exist to building classification trees and use different answers to the questions on tree construction (e.g., Kass (1980)). For example, it is possible to build trees with more than pairwise partitions at the nodes and to consider trees that ask more complicated questions involving more than one variable Brown & Pittard (1993).

Although trees have obvious interpretability advantages over other methods, they often suffer from less accuracy. Two of the more important recent extensions are boosting Freund & Schapire (1997) and random forests Breiman (2001). Boosting provides a method for trees to improve in accuracy by adapting to the errors they make in classification.

Random forests provides a mechanism for combining results from multiple classification trees to produce more accurate predictions. The random forests (RF) algorithm grows a group of classification trees (a forest). The RF algorithm constructs each tree in the forest using a

modified version of the recursive partitioning algorithm. One modification the RF algorithm makes is that it construct the trees using a subset of the data drawn from the original data set by sampling with replacement. This is known as "bagging".
The RF algorithm also modified the choice of questions procedure. For each question the RF algorithm considers only a subset of the available variables. The RF algorithm chooses this subset at the begging of the tree growing process and keeps it constant throughout tree growing.
Finally, RF has modifies the labeling or estimation of the response. Since we now have a forest rather than a single tree the label provided by a leaf node containing an observation in one tree may differ from the leaf node containing that same observation in another tree. For a categorical response the RF algorithm labels a new observation by a vote among the leaf nodes containing the observation in all trees. For a numeric response the RF algorithm estimates the value as the mean or median of the values produced by the respective leaf nodes in all trees.
Random forests sacrifices the interpretability of a single tree for the improved accuracy provided by an involved sampling and merging scheme. The RF approach recovers some of the interpretability by constructing forests with changes to the original data set. These changes involve sampling without replacement the values of a single variable using data not used in the original forest construction (i.e., the "out-of-bag" data). The difference in performance of the forest with the newly sampled variable and the original variable values gives a measure of importance for that variable. However, we do not recover the relationships among variables. The RF algorithm does provide a rough measure of interactions by finding the number trees with commonly paired variables and comparing this to random pairing. Comments on this procedure are in Breiman (2001).

3.3 Evaluation

Unlike unsupervised learning techniques, we can and should evaluate results from supervised learning techniques. Evaluation requires testing procedures and metrics. The goal of testing procedures is to provide an objective view of the performance of the unsupervised learning technique on future observations. For many reasons it is best not to rely on the observations in the database that were used to parameterize a technique to assess its performance on future values. The major reason for this caveat is because each technique can be made to perform perfectly on a set of observations. However, this perfect performance on a known data set would not translate into perfect performance on newly obtained observations. In fact, the performance on these would be quite poor because we *overfit* the technique to the existing data set.
Testing procedures provide a way to avoid overfitting. The simplest testing procedure is to divide the database into two parts. One part, the training set, is used to build and parameterize the data mining technique. The second part is used to test the technique. For reasonably sized databases the division is normally two thirds for training and one third for testing. In addition, the choice of observations for each set is randomly made. It may be useful to use stratified sampling for either of both of the training and test sets if the distributions of groups within a target population is known.
Cross validation is another testing procedure that is used when the database is small or when concerns exist about the representativeness of a test set. Cross validation begins by dividing the data into M roughly equal sized parts. For each part, $i = 1, \ldots, M$ the model is fit using the data in the other $M - 1$ parts. The metric is then computed using the data in the remaining part. This is done M times giving M separate estimates of the metric. The final estimate for

the metric is simply the average over all M estimates.
Cross validation has the advantage that it uses all the data for both training and testing. This means that the analyst does not have to form a separate test set. Recursive partitioning, discussed in Section 3.2 uses cross validation to determine the final size of the tree. In this way cross validation is frequently used to find parameter values for the different data mining techniques. For those methods that do not use it for parameter estimation it provides a convenient testing approach to assess a data mining technique.
In addition to testing procedures, the analyst must also select a metric or metrics to use to evaluate the techniques. For numeric response problems, common metrics are functions of sums of squares or sums of absolute deviations. Both measures weight performance by distance to the correct response, but the former measure tends to penalize extreme errors more than measures that use absolute deviation.
For categorical response, metrics that count the number of errors are typically used. However, in many applications the type of error is also important. This is particularly true in diagnostic applications. In these cases it is convenient to separate the errors into false positives and false negatives. False positives occur when the data mining technique predicts an outcome and the outcome does not occur. False negatives happen when the data mining technique fails to predict an outcome that occurred. The diabetes example illustrates a case where these two errors are not equally weighted. In this a case a false negative typically is worse than a false positive since the latter error can be caught by subsequent testing. On the other hand, it would be disastrous if only false positives occurred since this would quickly overwhelm the available testing resources. Hence, in performing evaluations on classifiers both types of errors need to measured and trade-offs made between their predicted values.
A useful display that allows for viewing of both metrics is the Receiver Operating Characteristic (ROC) curve. The name for this graphic derives from its origin in WWII where it was used by the allies to assess the performance of early radar systems. The ROC curse shows the trade-offs between false positives and false negatives by plotting true positives (1-false negatives) versus false positives. This means that the ideal performance is in the upper left hand corner of the plot. The worst performance is in the lower right hand corner. Random performance is shown by a diagonal line at 45°.
ROC curves often show there is no one, clear winner among the techniques. This happens frequently because the lines in the ROC curve cross (this will be illustrated in Figure 4 in Section 5). The choice in these cases become a matter of trade-offs between false positives and false negatives.

4. Integrated learning for problem discovery

Returning to the requirements for problem discovery we described in Section 1, we noted the need for techniques that provide

1. Data association;
2. Outlier identification and exploitation;
3. Interpretable relationships; and
4. Integrate operation.

The elements of this section provide for each of these capabilities by filling in the gaps in existing techniques noted in the previous two sections. Subsection 4.1 describes unsupervised learning methods for data association. Subsection 3 describes the use of supervised learning

methods with the interpretability needed for problem discovery. We combine all of these methods into a useful package for problem discovery in Subsection 4.3.

4.1 Data association

Data association refers to techniques that can find patterns that represent consistent causal or association mechanisms among the observations given evidence in the measured variables. To accomplish this goal data association uses and extends clustering and variable association to help uncover the mechanisms behind the problems in the domain of interest.

As with clustering (see Section 2.3) data association begins with measures of distance, similarity, or dissimilarity. To simplify our discussion here we consider only similarity. However, unlike general clustering algorithms, data association uses similarity measures tailored to the problem domain. Our approach to data association computes a problem tailored measure called a total similarity measure (TSM) between observations, $x_j, x_k, j, k \in \{1, \ldots, N\}$. The TSM is tailored to the domain through as a weighted composition of individual variable similarities or

$$\mathrm{TSM}(x_j, x_k) = \frac{\sum_{i=1}^{p} w_i \alpha_i(x_{ji}, x_{ki})}{\sum_{i=1}^{p} w_i} \tag{9}$$

where $w_i, i \in \{1, \ldots, p\}$ are the weighting coefficients and $\alpha(x_{ji}, x_{ki})$ are similarity scores for each variable,$i \in \{1, \ldots, p\}$. Both the weights and the variable similarities are scaled between zero and one, so for $i \in \{1, \ldots, p\}, j, k \in \{1, \ldots, N\}$

$$\begin{aligned} w_i &\in [0,1] \\ \sum_{i=1}^{p} w_i &= 1 \\ \alpha(x_{ji}, x_{ki}) &\in [0,1] \end{aligned} \tag{10}$$

The similarity measures are typically scaled differences for quantitative variables and partial match scores for binary variables. Details are in (Brown & Hagen (2003)).

To tailor the TSM to the problem domain the weights are adjusted based on the observed values. Values common across all observation do not provide as much information for problem discovery as those with greater diversity. Greater diversity means that the occurrence of the same values in multiple observations gives greater confidence that these observations have common causal or association mechanisms. We formalize this idea using information theory. Let $\mathcal{I}(x_j \sim x_k; x_{ji}, x_{ki})$ represent the information that observations x_i and x_j have the same causal or association mechanism given the values of variable k for both observations. Now consider the following axioms from information theory as applied to this data association problem.

1. $\mathcal{I}(x_j \sim x_k; x_{ji}, x_{ki})$ should be a function only of the prior probability of causality or association before the values of the variable k are obtained and only of the posterior probability after their measurement.
2. If the values of two variable are statistically independent evidence of the causality or association of the observations then the combined information in their measurement should be the sum of the information provided by their separate, sequential measurement. Formally,

$$\mathcal{I}(x_j \sim x_k; x_{ji}, x_{ki}, x_{j\ell}, x_{k\ell}) = \mathcal{I}(x_j \sim x_k; x_{ji}, x_{ki}) + \mathcal{I}(x_j \sim x_k, i; x_{j\ell}, x_{k\ell}). \tag{11}$$

The left hand quantity is the information about the causality or association of the observations when we get the values of both variables simultaneously. The first term on the right is the information we would get from first obtaining the values on one variable (i). The second term on the right is the information we would get from now updating the information we had from variable i with the arrival of the values of variable ℓ.

3. Finally we require the evidence for causality or association in multiple instances to be additive. For example, suppose we have four observations, a,b,c, and d. Then the information that x_a and x_b associate given the evidence in variable i plus the information that x_c and x_d associate given the evidence in variable ℓ should equal the information that they associate given the simultaneous presence of the information. Formally,

$$\mathcal{I}(x_a \sim x_b, x_c \sim x_d; x_{ai}, x_{bi}, x_{c\ell}, x_{d\ell}) = \mathcal{I}(x_a \sim x_b; x_{ai}, x_{bi}) + \mathcal{I}(x_c \sim x_d; x_{c\ell}, x_{d\ell}) \tag{12}$$

Taken together these axioms imply (see Feinstein (n.d.)) that information for causality or association given in the values of variable i for records j and k should be measured by

$$\mathcal{I}(x_j \sim x_k; x_{ji}, x_{ki}) = \mathcal{K} log\left(\frac{Pr(x_j \sim x_k; x_{ji}, x_{ki})}{Pr(x_j \sim x_k)}\right) \tag{13}$$

where $\mathcal{K}$ is a constant, the numerator is the posterior probability of mutual causality or association given the evidence in variable i, and the denominator is the prior probability of mutual causality or association.

Now since the observations may or may not have common causality or association, we want to measure the expected value of the information given the values measured for variable i. We take the expectation under the distribution for the posterior which gives a measure known as the Kullback-Leibler divergence or relative entropy (see, Brown & Smith (1990)):

$$\begin{aligned} \mathfrak{I}_i(x_j, x_k) \;=\; & Pr(x_j \sim x_k; x_{ji}, x_{ki}) log\left(\frac{Pr(x_j \sim x_k; x_{ji}, x_{ki})}{Pr(x_j \sim x_k)}\right) \\ & + Pr(x_j \nsim x_k; x_{ji}, x_{ki}) log\left(\frac{Pr(x_j \nsim x_k; x_{ji}, x_{ki})}{Pr(x_j \nsim x_k)}\right) \end{aligned} \tag{14}$$

where $x_j \nsim x_k$ indicates that the observations do not have a common causal or association mechanism. Notice that this measure treats variables that give negative evidence about causality or association in the same way as positive evidence. Taken together equations 13 and 14 provide metrics for the information found in the value of a variable. In other words, a metric that dynamically adapts to the specifics of data association for problem discovery. This dynamic metric defines the weights, $w_i, i = 1, \ldots, p$ for the variables in equation 9

To use this metric we need to estimate the prior and posterior probabilities. These can be found from the observed frequencies in the data (see Brown & Hagen (2003)). The new Total Similarity Measure (TSM) with the information theoretic based weights is

$$\text{TSM}(x_j, x_k) = \frac{\sum_{i=1}^{p} \mathfrak{I}_i(x_j, x_k) v_i \alpha_i(x_{ji}, x_{ki})}{\sum_{i=1}^{p} \mathfrak{I}_i(x_j, x_k) v_i} \tag{15}$$

where the $v_i, i = 1, \ldots, p$ ensure satisfaction of the conditions in (10).

When used with clustering algorithms, such as those described in Section 2.3, the TSM in (15) provides us with a way to identify groups of observations with possible common causal or association mechanisms. However, the approach described in this section applies only to fixed field data or variables with defined levels. For free text or narrative variables we need results from text mining. The next section explores our approach to incorporating free text for problem discovery.

4.2 Text association

As discussed in Section 1 text association is a critical component of problem discovery. This follows from the common occurrence of text in domains that have interesting but complex causal and association relationships. For example, understanding the causal or association factors behind accidents, medical conditions, and even customer behavior requires the incorporation of evidence from text to fully understand the complex relationships in these domains.

As indicated in Section 2.4 A major problem with existing text mining and natural language processing is the computational complexity of the methods. This limits their usefulness for problem discovery where the amount text can overwhelm many current techniques. Additionally, methods from information retrieval require query specification to get the documents related to the query. Also information retrieval techniques compute the similarities between documents based only on the similarities between terms in the query and terms in the documents. Hence, high similarity scores between terms do not imply causation or association if the query was not well chosen.

To measure the similarities between narratives, we describe the use of High Information Content Words (HICW) to represent the narratives. We compute similarities between HICW as surrogates for similarities between narratives. We begin our description with a brief introduction to HICW and follow this with our method for using HICW for computing similarities between text and narratives.

High Information Content Words are a set of words selected from an observation's narrative that provide important information for distinguishing the observation and determining its similarity to other observations with possibly identical causal or association mechanisms. HICW have two features: the ability to represent the narrative and the ability to distinguish the observation.

HICW is not the same as the keywords of the observational narratives. Keywords are a set of words which can summarize the narratives. Although keywords can represent narratives, they may lose important information about the observations needed to understand causal or association mechanisms. For example, suppose we have a collection of narratives about accidents. One of the narratives states "An derailment occurred when a southbound passenger train struck a maintenance vehicle on the track." Another narrative states "A head-on collision and subsequent derailment occurred when an eastbound freight train failed to change tracks and struck a westbound freight train. The engineer of the eastbound train tested positive for drug use." A keyword for both narratives would be "derailment", because both narratives describe derailment accidents and this one word provides a nice summary. However, these two observations have different HICW. Derailments may occur with sufficient frequency that this word would not distinguish these observations from other observations. More importantly this word does not help capture the causal or association mechanisms. For this we need words like "maintenance vehicle on track" and "drug use." Hence, using HICW we seek to find these factors that can help with problem discovery.

HICW also provide a computational advantageous approach to measuring similarities between narratives. Rather than compute the similarities using all words in the narrative, the HICW approach focuses on a small but informative set of words. Perhaps more importantly, HICW excludes words with limited information values.

In order to use HICW for text association, we first generate a word dictionary. This word dictionary derives from the corpus of all narratives within the observations. The word dictionary lists words and their Inverse Document Frequency (IDF). IDF for word i is calculated as

$$\text{IDF}_i = log_2\left(\frac{N}{n_i}\right) \tag{16}$$

where N is again the number of observations but also the number of narratives and n_i is the number of narrative that contain word i. Importantly, only content words are included in the word dictionary. Content words include nouns, verbs, adjectives, and adverbs, but exclude articles, conjunctions, and pronouns.

To generate HICW from a narrative, we first measure the importance of each word in the narrative. The importance of a word is decided by two criteria: the ability to represent the narrative and the ability to distinguish the observation. To paraphrase (Salton (n.d.)), the more times a word occurs in a narrative, the more likely the narrative is about this word and the greater the number of narratives containing the word, the less distinctively the word describes any of those narratives. Therefore, we can measure the importance or weight, w_{ij}, of a word i in narrative j by

$$w_{ij} = \text{TF}_{ij} \times \text{IDF}_i \tag{17}$$

where IDF_i is given in equation (16) and TF_{ij} is the term frequency of word i in narrative j defined as follows:

$$\text{TF}_{ij} = \frac{\text{tf}_{ij}}{\max_j\{\text{tf}_{ij}\}}. \tag{18}$$

To generate the HICW from a narrative, the importance of each word in the narrative is computed using equation (17). Next the words are ranked based on these importance scores. A specified number of words with the highest importance are the HICW. Clearly if the number is too small we do not capture possibly important characteristics of the observation. On the other hand, if the number is too large we increase computation time and risk including insignificant words. We have found using test sets or cross-validation (see Section 3.3) provide good selection criteria for this number.

Once we have the HICW we can compute the similarities between narratives, and hence, the observations that contain those narratives. The simplest method to measure this similarity, S_{ij}, between narratives i and j is to compute

$$S_{ij} = \frac{2M_{ij}}{m_i + m_j} \tag{19}$$

where m_i and m_j are the number of words narratives i and j, respectively. M_{ij} is the number of words they have in common.

We can also measure this similarity using a synonym dictionary. This gives the following similarity measure

$$S_{ij} = \min\left\{ \frac{2\sum_{k=1}^{m_i}\sum_{\ell=1}^{m_j} \mathrm{Sy}(W_{ki}, W_{\ell j})}{m_i + m_j}, 1 \right\} \tag{20}$$

where W_{ki} and $W_{\ell j}$ are the k^{th} and ℓ^{th} HICW in narratives i and j, respectively. Sy() is the synonym dictionary function which returns a value for the synonym match between the words given as inputs. At its simplest this is a binary function that indicates whether the words are synonyms. A more sophisticated synonym function returns a value between zero and one indicating the quality of the synonymy.

4.3 Supervised learning for problem discovery

Recalling the goal for problem discovery we want to find patterns or relationships that indicate causal or association mechanisms. As canonical examples of problem discovery we have used the discovery of factors causing or contributing to accidents or disease states. As we noted in Sections 2 and Section 3 data mining techniques provide a foundation for this work, but they cannot answer the problem questions in isolation from each other.

We have found an integrated approach to problem discovery that marries the results from unsupervised learning with supervised learning works well. This approach has the following steps:

1. Calculate the similarity between observations using the adaptive techniques described in Sections 4.1 and 4.2;
2. Cluster the observations using one or more the clustering techniques described in Section 2.3;
3. Use interpretable supervised learning techniques, such as those described in Section 3 to validate the cluster solution or solutions; and
4. If validated, use the insights provided by the interpretable supervised learning techniques combined with the structure identified by the clustering procedures to identify causal or association mechanisms.

The previous sections have provided an overview to the conduct of steps 1 – 2. In this section we turn to the final steps, 3 – 4 in our integrated method.

The results from the unsupervised learning step will yield a clustering solution that we can represent as a probability density function over the space of variables. For example, equation (5) shows this density function as a mixture of Gaussian densities. Using the insightful approach of Breiman (Breiman (2001)) we can apply supervised learning to help validate the cluster solution.

To apply this approach, let $f_(x)$ represent the density given by the original data. It is this density that our cluster solution has estimated. We now take independent draws from each variable in the data set and call this new distribution $f_I(x)$. Notice that if the original data set contains structure in $f(x)$ which we approximated with our cluster solution, then the independent variable distribution, $f_I(x)$ contains none of this structure. Thus, we can treat the observations in the original data set as coming from class 1 and the observations created by the independently sampled data set as class 2. This formulation enables the use of supervised learning to indicate the separability of the observations from the two classes. The greater the accuracy of supervised learning methods on test sets drawn from these two distributions the more confident we are in the clustering solution, and hence, the patterns this solutions provides. The lower accuracy reduces our confidence in the clustering solution.

This use of supervised learning can provide more general comparisons as indicated in Hastie et al. (2001). Instead of forming the distribution $f_I(x)$ using independent and uniform draws on each of the variables, we can instead create the new distribution according to any appropriate reference distribution. This enables a richer set of comparisons with the distribution observed in the original data set. To this we again apply supervised learning to classify observations from $f(x)$ and $f_I(x)$. Again we are interested in using the supervised learning methods to give us measures of departure of the original distribution from the reference distribution.

If the application of supervised learning shows significant departure from the reference distribution (say, with classification accuracy of better than 70%) then we can proceed to further understand the characteristics and relationships in the problem domain. For instance, in addition to providing a validation measure of the clustering solution, when this supervised learning approach is implemented with the methods described in Section 3 it can reveal variable importance, the presence of outliers, and variable interactions and nonlinearities. These characteristics can help narrow the focus of the problem discovery and allow for variable reductions or shrinkage (using the methods in Section 2.2). Once the number of variables is reduced or the variables are transformed using principal components or singular value decomposition, we can find a new clustering solution. Again we apply supervised learning to this clustering solution and repeat this process until the change is minimal (below some predefined threshold).

The previous two sections have provided us with the similarity measures that can be directly tailored to the specifics of the problem discovery domain of interest. Section 4.1 showed an information theoretic formulation for adaptively weighting and then scoring the similarities of values from fixed field variables. Section 4.2 showed how we can get adaptive weights and similarity scores for free-text variables. Once we have these similarity scores we can use any of the clustering techniques.

A third requirement for problem discovery methods applies to the use of supervised learning techniques. Specifically these techniques must produce interpretable results. This means that the discovery methods must reveal insights into causal or association mechanisms that contribute to the problem. So, unlike traditional data mining, problem discovery focuses more on interpretability at the possible expense of accuracy.

Finally, problem discovery requires the integration of methods from both supervised and unsupervised learning. By definition the exact nature of the problem is unknown so the application of, say, supervised learning tends to provide a narrow focus that misses important aspects of the problem. In contrast, unsupervised learning provides too broad a perspective in the presence of known instances of problematic behavior. Hence, problem discovery requires integrated strategies that combine results from supervised and unsupervised learning approaches.

5. Problem discovery example

As an example of the of the method described in the previous sections for the problem discovery, consider the rail operations in the U.S. In particular, the U.S. wants to reduce the number and severity of train accidents. Positive Train Control (PTC) has been advocated as an approach to enabling the desired reduction. PTC consists of a suite of technologies, e.g., accelerometers, controllers, temperature, humidity, and other environmental sensors, and GPS. The Federal Railroad Administration (FRA) has spent more than 15 years in development of PTC and expects to deploy this technology later this decade Administration

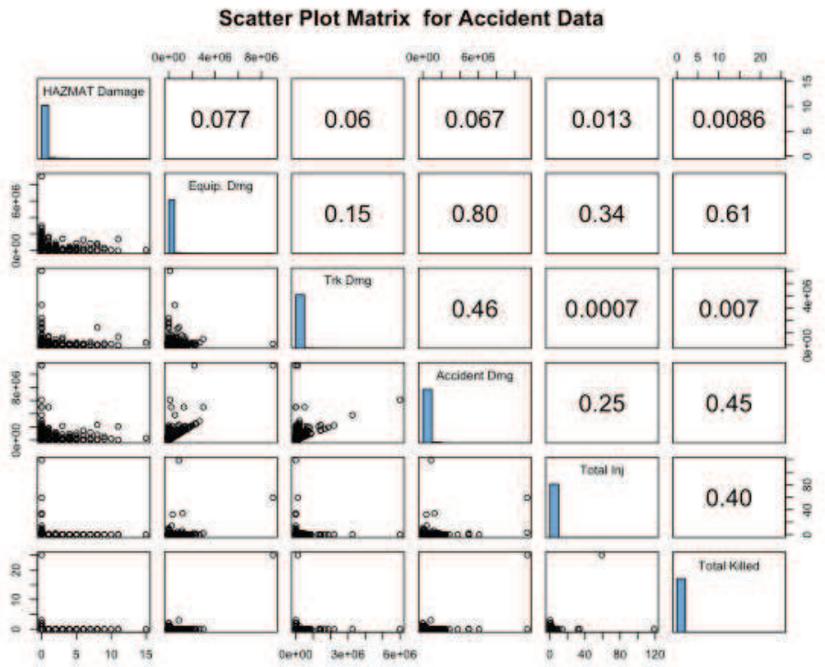


Fig. 2. Scatter plot and correlations for accident outcome variables.

(2009b). "The National Transportation Safety Board (NTSB) has named PTC as one of its "most-wanted" initiatives for national transportation safety" Administration (2009b). Beginning in 2001 the railroads deployed components of PTC on small sections of track to test and validate its usefulness. A complete list of these deployments is in Administration (2009b).

Despite the development and incremental deployment of this technology, rail operators in the U.S. do not fully understand the causes or associated mechanisms behind train accidents. They specifically do not know how the number and severity of these accidents will be affected by the deployment of PTC.

To apply problem discovery methods to train accidents we use the data available on accidents for the last decade Administration (2009a). The data consist of yearly reports of accidents and each yearly set has has 141 variables. The variables are a combination of numeric, e.g., accident speed, categorical, e.g., equipment type, and free text. The free text is contained in narrative fields that describe the accident. We can divide the fixed field variables into three categories: control, exogenous, and outcome. The control variables, such as, speed can be set by the engineer or train operator. The exogenous variables like weather provide uncontrollable conditions at the time of the accident. Outcome variables measure the results of the accident. Examples of these results are the cost of damage and the number of people injured or killed.

The train accident data are typical of other types of accident data in that they are highly skewed. Figure 2 shows pairwise scatter plots and linear correlations among the outcome variables. This figure shows that most accidents have little damage or loss of life. Extreme events do occur, however. The question for problem discovery methods is to find patterns in these events that may guide solutions.

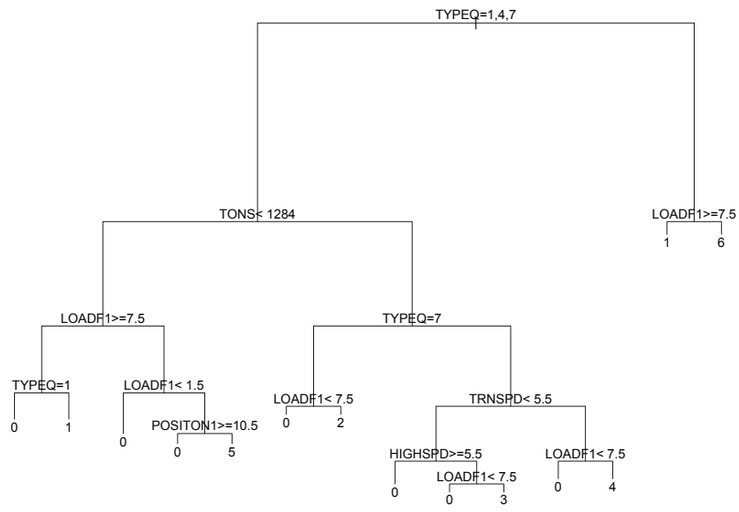


Fig. 3. Tree representation of clusters found by data association.

Applying the methods discussed in Section 4 we first apply clustering techniques. In this case we applied the data association and the text mining methods from Sections 4.1 and 4.2. We used both nearest neighbor and model-based clustering algorithms with the similarity scores computed as those sections describe. The results can be viewed in several ways. Figure 3 shows a classification tree representation of the clusters found with just the fixed field variables. This representation is convenient, but somewhat misleading since other variables become important to the relationship as we include the high information content words (HICW) from the narratives. Nonetheless, it does show the use of data association to uncover patterns in the data.

Before proceeding with additional problem discovery we need to validate that the data contain enough structure to justify the clustering results. Figure 4 shows the ROC curves from applying both random forests and recursive partitioning to the data and to two data sets randomly created with the variables and values given in the original data. These two new data sets are random permutations of the original values. Then using the method described in Section 4.3 we sought to discover if the original data could be accurately discriminated from the random sets. To make this comparison we built the models using approximately two thirds of the data and tested with the remaining one third. As this figure shows, both random forests and recursive partitioning provide highly accurate models on these out-of-sample data. This suggests that the data do contain relationships susceptible to problem discovery.

With those results we applied the combined data and text association techniques described in Sections 4.1 and 4.2. Figure 5 shows the four cluster solution projected into the first two principal components of the outcome variables. Principal components are described in Section 2.2. This solution suggests that there are some causes that have particular relevance to the outcome variables. As an illustration of high information words, the HICW found that accompany cluster 1 are “drugs”, “alcohol”, and “positive.”

Also of interest are indicators of variable importance. Figure 6 shows the variable importance

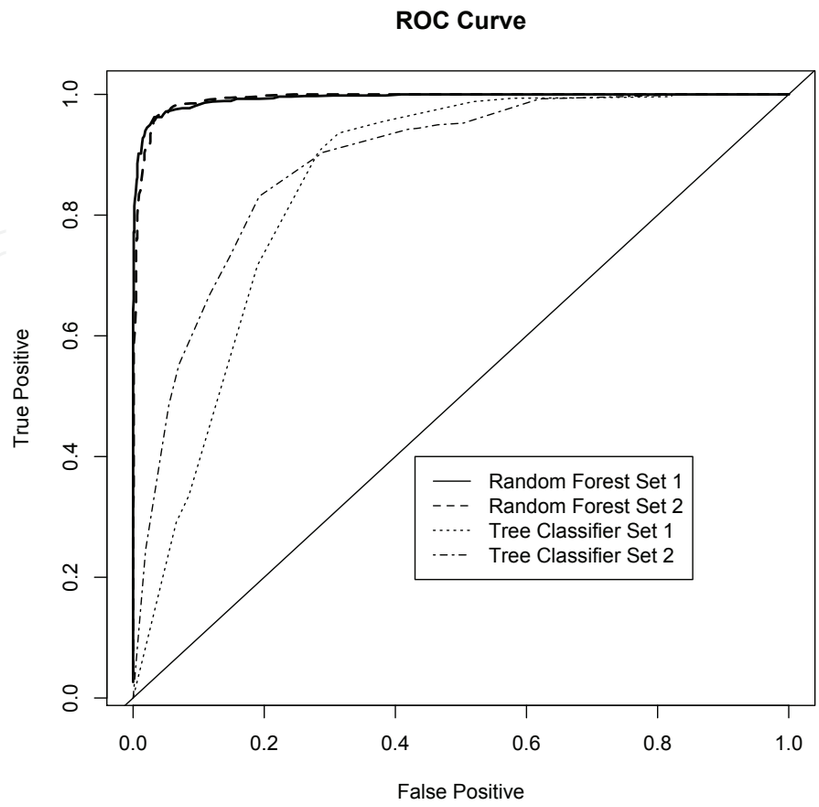


Fig. 4. ROC validation plots for cluster structure in the data.

plot found by apply random forests to the discovered types of accidents. The variables suggest the components that contribute to each of the different types of accidents discovered in the data and text association. This variable importance applies only to the fixed field variables.

The results from this problem discovery exercise suggest that most accidents will not be affected by the use of PTC. Further the most extreme accident was a head-on collision with an HICW of "drugs". This incident had a total cost of $11M, 25 killed, and 62 injured. It clustered with others accidents that were not as costly but nonetheless were more damaging the median. While the potential for head-on collisions can be detected by PTC it is not clear that PTC would matter given mental state train operator. The largest cluster of accidents had little cost and very low speeds. An HICW for accidents in this large cluster is "fouling," and this will continue to occur with the same regularity and cost even if PTC is fully implemented. Another cluster that had deaths or injuries greater than zero concerned crossing and intersection accidents. These would not be affected significantly by PTC. However, there are a small number of accidents at speeds and conditions that suggest that PTC could have an influence. Unfortunately removing them will not greatly impact the overall severity of accidents.

Clearly this exercise suggests that investment in other strategies may produce more significant results for reducing the number and severity than PTC. For instance, warning systems for equipment on the tracks or at grade crossing will have the most effect on reducing the number of those killed by trains.

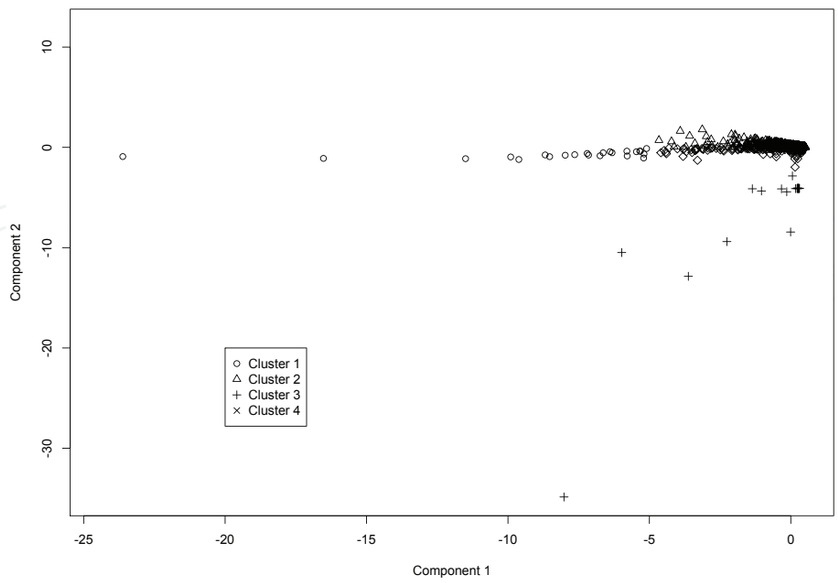


Fig. 5. Clusters showing types of accidents as projected into the first two principal components.

From the perspective of this chapter, this exercise shows the effectiveness of combined methods from unsupervised and supervised learning.

6. Conclusion and future work

Problem discovery represents a major application for data mining techniques. The goal of problem discovery is to find causal or association mechanisms and the discovery processes in data mining can contribute greatly the achievement of this goal. However, to make this happen requires that data mining techniques address the four key requirements of problem discovery: data association; text association; supervised learning for structural characterization; and integration of methods.

While unsupervised learning has techniques and methods similar to those need in the areas of data and text association, we note that there are gaps. The methods described in Sections 4.1 and 4.2 show ways to fill these gaps. For data association Section 4.1 describes the use of information theory to obtain similarity measures tailored for problem discovery. Section 4.2 illustrates how high information content words relevant to the causal and association mechanisms can be found and exploited.

Once we have the initial clustering structure evident in the data, we can apply supervised learning to provide greater insights into the nature of this structure. As Section {subsec:int shows, supervised learning also provides the means for validating the structures found by the unsupervised learning methods. These results then lead to another round of unsupervised learning and closer inspection of the variables indicated as important by the supervised learning techniques. Section 4.3 also show the general integration methodology for coupling the supervised and unsupervised techniques.

Finally, Section 5 provides an example of the use of these techniques to understand the problems at the foundation train accidents in the U.S. In so doing, it provides an critique of the

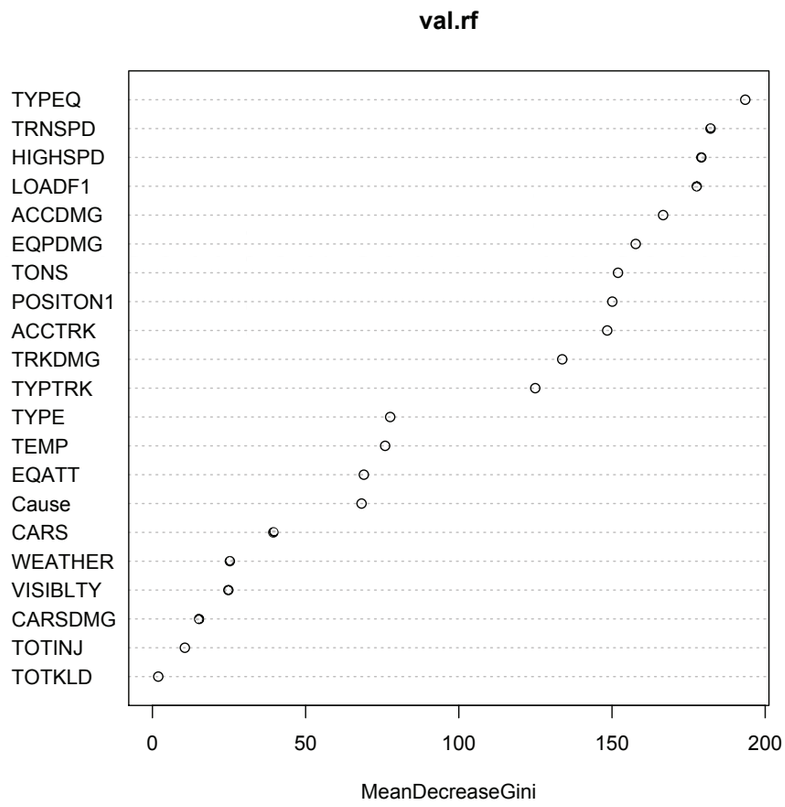


Fig. 6. Variable importance in classifying the types of accidents

pursuit of technologies such as Positive Train Control as the means to reduce the extent and severity of accidents. The section shows the results from the data and text association. It also shows the structures discovered by supervised learning. Finally, it shows how this integrated approach to problem discovery can guide designs for addressing the factors most relevant to accomplishing the goal of the rail operators to diminish the number and costs of accidents.

The methods presented here show promise for improving problem discovery. However, many important challenges remain to extend methods such as these across a wider range of applications. First, methods are needed to incorporate the temporal characteristics. Temporal data are correlated and this correlation structure needs to be well-modeled if we are to understand the problem mechanisms related to time versus other causes.

Similarly, spatial characteristics should be specifically modeled as part of the problem discovery toolkit of techniques. As with time, spatial variables have special correlation structures. These structures require methods more directed than the overarching approaches currently used, particularly in unsupervised learning.

Finally, the integration of supervised and unsupervised learning methods for areas like problem discovery is not well studied. Unlike the combination of supervised learning techniques, which has received considerable attention, the integration of methods from both general areas remains a matter of folklore rather than rigorous investigation. This has to change. Problem discovery requires richer integration of these methodological areas to provide techniques for improving our understanding of the possibly complex relationships

among variables at the heart of causal and association mechanisms. This is true for more than problem understanding where the need is particularly evident.
Overall problem discovery will grow in importance as the challenges of dealing with complex issues in health care, energy, transportation, and other areas become evident and pressing. The methods described in this chapter introduce the critical use of ideas from data mining to aid in the problem discovery process. If we are successful the next decade will witness major advances in this important field.

7. References

Administration, F. R. (2009a). Office of safety analysis. http://safetydata.fra.dot.gov/officeofsafety/.

Administration, F. R. (2009b). Positive train control (ptc). http://www.fra.dot.gov/us/content/784.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, A. (1996). Fast discovery of association rules, *in* U. Fayyad, G. Pietsky-Shapiro, P. Smyth & R. Uthurusamy (eds), *Advances in Knowledge Discovery and Data Mining*, AAAI/MIT Press, Cambridge, MA, pp. 307–328.

Baluja, S., Mittal, V. & Sukthankar, R. (1999). Applying machine learning for high performance named-entity extraction, *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, Pacific Association for Computational Linguistics, Watterloo, Canada.

Bishop, C. (1999a). Bayesian pca, *in* M. Kearns, S. Solla & D. Cohn (eds), *Advances in Neural Information Processing Systems, Volume 11*, MIT Press, Cambridge, MA, pp. 382–388.

Bishop, C. (1999b). Variational methods in principal components, *Proceedings of the Ninth International Conference on Artificial Neural Networks, ICANN*, Vol. 1, IEE, pp. 509–514.

Breiman, L. (2001). Random forests, *Machine Learning* 45: 5–32.

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*, Wadsworth, Belmont, CA.

Brown, D. E. & Hagen, S. (2003). Data association methods with application to law enforcement, *Decision Support Systems* 34: 369–378.

Brown, D. E. & Smith, R. L. (1990). A correspondence principle for relative entropy minimization, *Naval Research Logistics* 37: 191–202.

Brown, D. & Pittard, C. (1993). Classification trees with optimal multi-variate splits, pp. 475–478. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, Le Touquet, France.

Ciravegna, F., Lavelli, A., Mana, N., Matiasek, J., Gilardoni, L., Mazza, S., Black, W. & Rinaldi, F. (1999). Facile: Classifying texts integrating pattern matching and information extraction, *Proceedings of 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*, AAAI, San Francisco, CA, pp. 890–895.

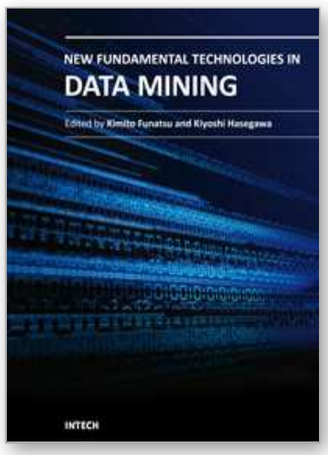
Comon, P. (1994). Independent component analsysis, a new concept?, *Technometrics* 36: 287–314.

Copas, J. (1983). Regression, prediction and shrikage (with discussion), *Journal of the Royal Statistical Society, Series B Methodological* 45: 31–354.

Corley, C. & Mihalcea, R. (2005). Measuring the semantic similarity of texts, *Proceeding of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, MI, pp. 13–18.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. (1990). Indexing by latent

semantic analysis, *Journal of the American Society of Information Science* 41: 391–407.
Fall, C., Torcsvari, A., Benzineb, K. & Karetka, G. (2003). Automated categorization in the international patent classification, *ACM SIGIR Forum* 37: 10–25.
Feinstein, A. (n.d.). *Foundations of Information Theory*, McGraw-Hill, New York.
Freund, Y. & Schapire, R. (1997). A decision theoretic generalization of online learning and an application to boosting, *Journal of Computer and System Sciences* 55: 119–139.
Gentili, G., Marinilli, M., Micarelli, A. & Sciarrone, F. (2001). Text categorization in an intelligent agent for filtering information on the web, *International Journal of Pattern Recognition and Artificial Intelligence* 15: 527–549.
Golub, G. & Loan, C. V. (1983). *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD.
Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The elements of statistical learning*, Springer Verlag, New York, NY.
Hoang, A. (2004). Information retrieval with principal components, *Proceeding of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, Vol. 1, IEEE Computer Society, Las Vegas, NV, p. 262.
Hoerl, A. & Kennard, R. (1964). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 12: 55–67.
Hotho, A., Staab, S. & Maedche, A. (2001). Onthology-based text clustering, *Proceedings of the IJCAI-2001 Workshop Text Mining: Beyond Supervision*, Springer-Verlag, Seattle, WA, pp. 264–278.
Kass, G. (1980). An exploratory technique for investigating large quantities of categorical data, *Applied Statistics* 29: 119–127.
Kedem, B. & Fokianos, K. (2003). *Regression Models for Time Series Analysis*, John Wiley and Sons, Inc., Hoboken, NJ.
Konchady, M. (2006). *Text Mining Application Programming*, Charles River Media, Boston, MA.
Krupka, G. R. & Hausman, K. (1998). Isoquest inc.: Description of the netowltm extractor system as used for muc-7, *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, Science Applications International Corporation, 10260 Campus Pt. Dr., San Diego, CA.
Mani, I. & Maybury, M. (1999). *Advances in Automatic Text Summarization*, The MIT Press, Cambridge, MA.
Salton, G. (n.d.). *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, Massachusetts.
Seber, G. (1984). *Multivariate Observations*, John Wiley and Sons, Inc., New York.
Witten, I. H., Bray, Z., Mahoui, M. & Teahan, W. (1999). Using language models for generic entity extraction, *Proceedings of the International Conference on Machine Learning (ICML 1999), Workshop on Text Mining*, Morgan Kaufmann, Bled, Slovenia.
Wold, H. (1975). Soft modeling by latent variables: The nonlinear iterative partial least squares (nipals) approach, *Perspectives in Probability and Statistics, In Honor of M.S. Bartlett* pp. 117–144.
Yetisgen-Yildiz, M. & Pratt, W. (2006). Using statistical and knowledge-based approaches for literature-based discovery, *Journal of Biomedical Informatics* 39: 600–611.
Zheng, Z., Kohavi, R. & Mason, L. (2001). Real world performance of association rule algorithms, *in* F. Provost & R. Srikant (eds), *Proceedings of the ACM SIGKDD Intentional Conference on Knowledge Discovery in Databases and Data Mining (KDD-01)*, ACM Press, pp. 401–406.



New Fundamental Technologies in Data Mining

Edited by Prof. Kimito Funatsu

ISBN 978-953-307-547-1
Hard cover, 584 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

The progress of data mining technology and large public popularity establish a need for a comprehensive text on the subject. The series of books entitled by "Data Mining" address the need by presenting in-depth description of novel mining algorithms and many useful applications. In addition to understanding each section deeply, the two books present useful hints and strategies to solving problems in the following chapters. The contributing authors have highlighted many future research directions that will foster multi-disciplinary collaborations and hence will lead to significant development in the field of data mining.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Donald E. Brown (2011). Data Mining for Problem Discovery, New Fundamental Technologies in Data Mining, Prof. Kimito Funatsu (Ed.), ISBN: 978-953-307-547-1, InTech, Available from: http://www.intechopen.com/books/new-fundamental-technologies-in-data-mining/data-mining-for-problem-discovery

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen